Barton, Alan J.

Blue Kenue enhancements from 2014 to 2019

Blue Kenue enhancements from 2014 to 2019

Alan J. Barton
Ocean, Coastal and River Engineering (NRC-OCRE)
National Research Council Canada (NRC)
Ottawa, Ontario, Canada
alan.barton@nrc-cnrc.gc.ca

***Abstract*—Blue Kenue® has been under development for about 20 years, and provides a framework for pre-processing, post-processing and visualization of hydrodynamic model data. The National Research Council Canada makes this software freely available for use by the open TELEMAC-MASCARET user community and benefits significantly from the knowledge that the community shares. The paper will briefly describe some of the key changes to Blue Kenue that have been implemented over the last 5 years (such as how attributes are improving, how internal organization is changing and which new features have been added) and will provide some initial thoughts about potential future Blue Kenue enhancements (such as improved data analytics and additional modelling support for ice-hydrodynamics, microplastics and/or oil). The paper is intended to help be a starting point to identify needs and priorities for future development.**

I. Introduction

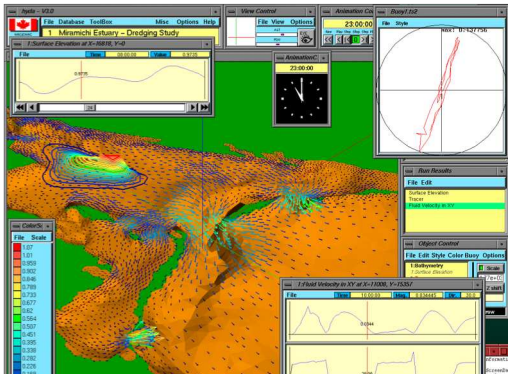
This paper briefly describes Blue Kenue's origin, the new Blue Kenue development environment, some ways that the new environment is helping inform the development efforts, and a few yearly examples of Blue Kenue additions since 2014 to help understand where Blue Kenue is now. The paper concludes by describing three possible Blue Kenue future development directions.

II. Blue Kenue Origin Story

Originally conceived by the Canadian Hydraulics Centre (CHC) of the National Research Council Canada in 1991, HYDA was designed to provide a single consistent user interface and database for collection, preparation, and analysis of hydro-numerical model data. Through use of powerful 3D visualization HYDA was designed to provide a real time virtual environment for modellers regardless of simulation methodology. HYDA's solver neutral database architecture allowed for easy interchange of data between different models. In 1994 CHC (then known as IMD's Hydraulics Laboratory) entered into a collaborative agreement to commercialize and further the development of HYDA technology. The result of this collaboration was the commercial HYDA version 3.2. Then, in 1997, CHC stopped supporting HYDA development in favour of the technology underlying Blue Kenue.

III. Blue Kenue Development Environment

Before 2014, Blue Kenue developers were using a version control system called *Microsoft Visual Source Safe*. In 2019, developers now rely on an online server called CHyMS; offering version control as one of its services.



Blue Kenue's conceptual predecessor was named HYDA

A. Canadian Hydrological Model Stewardship (CHyMS)

The original CHyMS server started to become operational in November 2011. In 2014, a major upgrade was performed that addressed a number of security related issues. The original CHyMS proposal can be found on CHyMS itself (in the Public Download Area). In essence, the proposal called for creation of a safe haven for Canadian hydrological models. A number of Canadian supporters were involved; including the Canadian Society for Hydrological Sciences. But how does CHyMS relate to Blue Kenue? Well, NRC also offers Green Kenue for free to the hydrological community. Therefore, Green Kenue developments and leading-edge installers were decided to be placed onto CHyMS for that community. And since the underlying technology behind Green Kenue overlaps with the underlying technology for Blue Kenue, both applications and the underlying technology made their way onto CHyMS. Therefore, CHyMS plays a key role in the development of Blue Kenue.

B. What services does CHyMS offer in 2019?

CHyMS has a Frequently Asked Questions (FAQ) webpage that is, co-incidentally, frequently updated. When read sequentially the FAQ offers answers to a broad number of questions, including what services are currently offered. In 2019, CHyMS offers the Blue Kenue community the Public Download Area for sharing. This sharing space is where NRC places its alpha and beta versions of Blue Kenue, among other things.

CHyMS also hosts both private and more open projects. Each project has been created with specific people in mind. Each project has tickets, version control, reporting, Kanban, and email reminders (if configured). In addition to per project

information, cross-project reporting is offered to a person for those projects in which they have been granted access. Such reporting helps inform decision makers.

CHyMS is also a place for communities to come together; such as the Green Kenue Community and the pyEnSim Community. There is also a Blue Kenue Community on CHyMS, however, that mainly directs people to the OpenTelemac forum [2]. These CHyMS communities will expand depending on the specific community preferences.

C. How do Blue Kenue developers use CHyMS in 2019?

There are 4 main CHyMS services that Blue Kenue developers use: (i) CHyMS Public Download Area, (ii) CHyMS Project Hosting, (iii) CHyMS Gentle Reminder Email Service, and (iv) CHyMS Kanban.

The CHyMS Public Download Area is a web page on CHyMS that provides links to files such as: fact sheets, executables, data files, tutorials, presentations, and videos amongst other things. These freely shared items are created by NRC and by other people and organizations that are also willing to share.

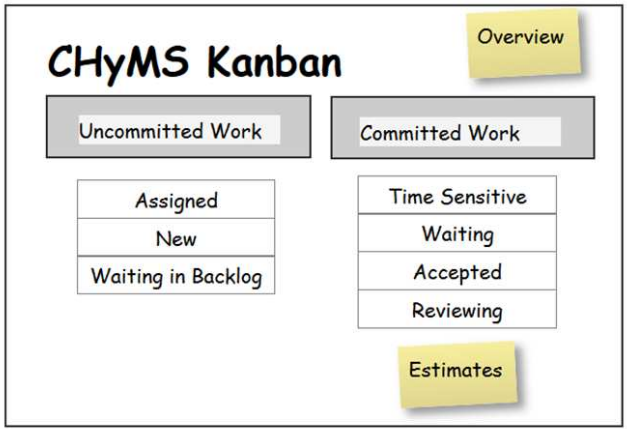
The CHyMS Project Hosting service offers a way for source code to be version controlled, tickets to be created, wiki pages to be shared, timelines to be viewed and other helpful things that most development projects might expect. One of the projects associated with Blue Kenue and hosted on CHyMS in 2019 is called "*EnSim Developer Stewardship Project*". This project is a large internally funded NRC project with specific goals and deliverables. A more detailed description follows in the next section.

The CHyMS Gentle Reminder Email Service is performed once per week. Each Monday morning CHyMS will collect all time-sensitive tickets across all hosted projects and will group the tickets by owner. CHyMS will then sort each person's owned tickets by due date and send the sorted time-sensitive information to the person. In this way, a person only receives one email for all time-sensitive tickets that they may have open. This helps to keep on top of work with a deadline.

The CHyMS Kanban is one way to visually represent a TODO list. This visual list typically comprises work, such as action items, bug fix descriptions, new features, etc. Each of these items is called a ticket. Such tickets may have comments such as status updates, attached files, images such as screen snapshots or links to other issues encountered. Each ticket also has attributes associated with it. These attributes can be customized. The CHyMS Kanban displays information about a few of the available ticket attributes and when a key attribute called "status" is changed, then the ticket automatically moves to another location in the Kanban. More details such as best practices for how to use the CHyMS Kanban are in the CHyMS FAQ.

IV. KNOWN WORK OVERVIEW

The EnSim Developer Stewardship Project is a large NRC-OCRE internally funded project that is primarily about refactoring the deepest part of Blue Kenue called EnSimCore. This part of Blue Kenue is a dynamic linked library (dll) that



The left (right) column shows tickets that have not been agreed (have been agreed) to be worked on. There are 8 locations for a ticket's information to appear (the 8[th] is closed). A ticket appears in a location based on its state; such as "Accepted". More details are available in the CHyMS FAQ [3].

is used by many other dlls and executables (exes) developed by NRC-OCRE. For example, both Blue Kenue and Green Kenue are executables that use the EnSimCore dll. The primary focus of the internal NRC-OCRE project is on: (i) improving attribute implementation, (ii) performing many code reviews to collect and describe potential errors, defects and faults, (iii) holding internal EnSim related discussions, and (iv) providing a small amount of time to fix a few external issues raised on the OpenTelemac forum. However, this internal project is not about building new features nor about addressing large amounts of technical debt left by other projects. Those things (and more) would be addressed under their own respective project(s).

Most of the known work is summarized (numerically) in the following table based on information stored on NRC's CHyMS server. Ideally, each piece of known work is fully described in its own ticket such that there is a one-to-one mapping between known work and CHyMS tickets. It is also important that each ticket is as unambiguous as possible so that completed work means that an open ticket may be closed. A review of all 394 tickets was performed in order to classify each ticket into one of four work directions. The 2 columns "open" and "closed" count (at a high-level) the number of tickets that are either still to be performed (open – new,

BLUE KENUE (AND OTHERS) KNOWN-WORK SUMMARY

Work Direction	**Counts for Tickets in Project on CHyMS [3]**			
	Open (Residual)	***Closed (Residual)***	***"Growth"***	***Calculated Strategy***
Strategic (large)	85 (-17.5)	23 (8.0)	-25.5	2
New Feature (small)	22 (45.5)	12 (19.0)	26.5	3
Proactive Maintenance	113 (-45.5)	38 (-7.0)	-38.5	**1**
Reactive Maintenance	48 (19.5)	46 (-15.0)	34.5	4
Total (394)	**270**	**124**		

assigned, backlogged, accepted, waiting, under review) or completed (closed – fixed, duplicate, will not fix, etc).

For example, from the table we can see that 48 tickets are open in the reactive maintenance work direction. In other words, there are 48 (smallish) things that people have noticed about Blue Kenue (or its technological relatives) that would be good to fix and/or modify. There are a total of 270 possible open work items to consider during the decision process of asking the question "*What should be worked on now?*" and a further 124 work items have been completed that do not need to be considered. In addition, a residual has been computed and placed in parenthesis in the table. For the reactive maintenance work direction, the residual is 19.5, meaning that this work direction has about 20 fewer tickets than the average work direction (there are 4 work directions in the table). In other words, all tickets having the same weight implies that this work direction is performing well as compared to the other work directions. Carrying the analysis of the residuals further, since we do not (yet) have trend information, we can see that the difference between the open and closed residuals may provide a sense of how each work direction has grown under the assumption that a ticket may belong to any of the 4 work directions with equal probability. A quick calculation may then provide an answer to the aforementioned question; and in our case, following the logic to its conclusion means that we should focus on the proactive work direction in order to lower the work in that area.

The flaw in this type of numerical analysis (automated or not), of course, is that all tickets are not equal weight for users of Blue Kenue and some users place more importance on some things than other users; depending on which task they are working. Some types of observed faults should certainly be resolved much more quickly compared against others. And some types of new features should not wait forever to get implemented. All of this to posit that funding, available developer time, developer expertise and shared user needs all *jointly* play a large role in the future success of Blue Kenue. So, let's all do our best to continue Blue Kenue's success.

One measure of success for Blue Kenue is whether the installer is continuing to be downloaded. The following table shows the download counts in the last 5 years and more. From the table, it seems like 2018 was a very prolific year; however, there may be some data error due various internal changes [6].

BLUE KENUE DOWNLOAD COUNTS (GRAND TOTAL: 15,044)

	Pre-	2014	2015	2016	2017	2018	2019[a]
[6] 32 bit	2537	352	273	303	197	159	87
[6] 64 bit	1868	1126	1196	1648	1524	2137	952
[3] 32 bit	-	-	-	5	30	53	8
[3] 64 bit	-	-	-	7	171	230	181
Total	4405	1478	1469	1963	1922	2579	1228

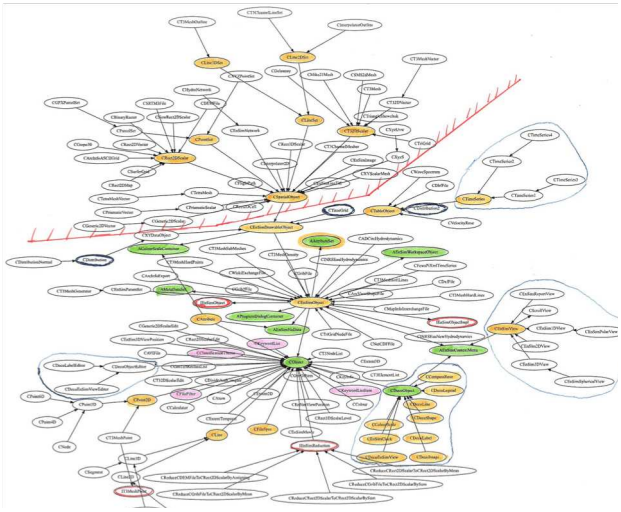
a. As of 16 August 2019

A more careful analysis would need to be carried out to improve the accuracy of that year's 32 bit and 64 bit counts (top 2 rows values). In any case, the general trend, as would be expected, is that 32 bit versions of Blue Kenue are not being downloaded as often as their 64 bit counterparts. What is surprising, however, is the fact that the 32 bit versions are not zero; there are still a couple of hundred downloads.

V. TECHNOLOGY OVERVIEW

Blue Kenue is built on top of a number of libraries; similar to Green Kenue, ECDE and others. The technology is mainly written in the C++ language (some parts are in Fortran and, exceptionally, pyEnSim allows the underlying technology to be exposed via the python language). This means that Blue Kenue needs to be fully compiled. Alternative language design approaches include, among others, interpreted languages, such as python, perl, or scripting languages such as bat, bash, bourne, etc., or interpret-compile hybrids, such as javascript. The underlying technology relies on the OpenGL library to provide all rendering functionality while application framework, menus, dialogs etc. are implemented via the Microsoft Foundation Class (MFC) library.

Creating an installer for Blue Kenue means that over 300,000 lines of code are selectively combined together in a meaningful way via a number of automatic and manual steps. Compilation means that one C++ source code file (*.cpp) is used to create one object file (*.o) and a set of object files are combined to make a static (*.lib) or dynamic (*.dll) library. Some object files and some of the static libraries are then linked together to make an executable (*.exe). The dlls are loaded into memory only when they are used at runtime. In other words, for example, when a menu item in Blue Kenue is



Example demonstrating some of the complexity of the system (not intended to be read in detail). Blue Kenue's C++ code is organized into classes (roughly 900). Each oval is one such class. Arrows go from one oval to another; signifying that one class is a child of another class. This figure shows only those parent/child relationships directly related to one very core class called *CEnSimObject*. In other words, this figure shows the connected component in which *CEnSimObject* belongs. Such a figure is helpful for developers learning the system architecture and for explanatory purposes.

used then the underlying implementing (if from a dll) means that the dll would be loaded into memory and then invoked. The executable for Blue Kenue uses about 8 static and 12 dynamic libraries. The installer (*.msi) is then made by combining the executable, the appropriate dynamic link libraries, the documentation, the changelogs with a description of new features, fixed bugs, experimental features, etc., and any required data files, such as base maps, databases, configuration files, etc. The resulting installer is also manually edited after construction in order to allow installation within NRC's new, and stricter, security environment. The installer is then placed onto CHyMS in the Public Download Area for sharing.

VI. Blue Kenue Examples

Blue Kenue enhancements occur when a project funds development effort. One such project is called the EnSim Developer Stewardship Project and is hosted by CHyMS [3]. This NRC-OCRE internally funded project has almost 400 tickets in total; of which about 100 are related to completed work. The following subsections describe an assortment of the tickets that were created in each of the years succeeding 2014 until the year 2019 and for which the associated work is now (mid 2019) complete. CHyMS was instrumental in helping to determine these examples through the creation of a wiki page grouping tickets by year.

A. Examples created in 2014 and now complete (9)

Ticket #48 – AVI recording fails on 64 bit. One comment on the ticket notes that the default encoder was Cinepak (that was not functioning properly) but that the other encoders do behave correctly. The fix was to generalize the existing code so that only those encoders that are installed on a user's machine (and for which Blue Kenue has implementations) will be presented within Blue Kenue for selection. This should stop situations where someone selects an encoder that they do not have installed and therefore for which they cannot record video. In other words, Blue Kenue has 4 implemented encoders and only a subset of those will be able to be selected from within the Recording tab of the 2D View properties. For example, on one development machine there are 7 unusable (by Blue Kenue) installed codecs and 3 usable (by Blue Kenue) installed codecs.

Ticket #65 – Time series tools not respecting missing data. In particular, computation of flow duration curves, cumulative sums, and integrals were reported to be not respecting the missing data value. Also of note, was the computation of distributions and performance statistics. The fix was to use a recently generalized *CAttributeSet* class which was used in many more places that described in this ticket. In other words, a lot of proactive maintenance was also involved in the resolution. The full and complete resolution touches many aspects of the 900 classes and is still a work in progress.

B. Examples created 2015 and now complete (5)

Ticket #78 – Cannot save points as (.pt2)*. In other words, creating a new XYZ point set using Blue Kenue resulted in creation of a new object in the workspace that could not be directly saved as *.pt2. During the investigation, it was determined that there is a work around; however, the addition of functionality to directly save to a *.pt2 file was added. A number of related issues were also identified; some of which were resolved.

Ticket #82 – Add double precision SELAFIN file support. A large effort was undertaken to add the required support. This included, among many other things, modifying display capabilities such as to add a "center of domain offset" and how the mouse interacts with it.

C. Examples created 2016 and now complete (5)

Ticket #103 – add "cell" drawing style to CRect2DScalar. A developer working on another application relying on parts of Blue Kenue's underlying technology submitted a patch file to add a new small feature. The patch was applied, reviewed, approved and committed to CHyMS. Subsequent Blue Kenue releases have included the submitted patch with the new display capability. In other words, a new drawing style has been added to Blue Kenue.

D. Examples created 2017 and now complete (4)

Ticket #106 – invalid y location for time series extraction. When extracting a time series at a point from a 2D spatial object (Rect 2D Scalar), the resulting metadata for the point's y location was incorrect. In addition, the same fix was made for 3D spatial object (Rect 3D Scalar) time series extraction.

*Ticket #114 – *.ts1 not properly displayed in 1D view*. Blue Kenue version 3.3.4 did not display the Time Series Type I file correctly. However, Blue Kenue version 3.9.5-beta displayed the time series as expected. In the future, the reporter of this fault will check the most recent version of Blue Kenue and only report a fault if it has not been fixed. Of course, this assumes that a person is aware of where to find the most recent version (Available in the CHyMS Public Download Area [3]).

E. Examples created 2018 and now complete (58)

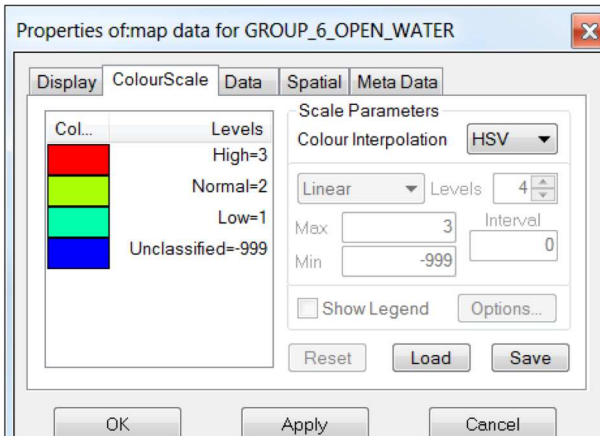
The approval of a new NRC-OCRE internal project has allowed almost 10 times as many tickets to be created during 2018. The resolution of which may have also occurred during 2019 along with other tickets that had been created in previous years. A few of the 58 tickets created in 2018 include:

Ticket #180 – Create AAttributeSet. Pull out the attribute related functionality from the class hierarchy and localize it. The reason is to provide more consistency in dealing with multiple attributes; including robustness improvements and in the long term, easing introduction of attribute-related new functionality along with some performance gains due to localized memory. Historically, this ticket was motivated by observed faults in a different application that, upon inspection resulting in the EnSim Developer Stewardship project being proposed to and funded by NRC-OCRE. In addition, historically, Blue Kenue only had one attribute and over time multiple attributes were slowly added on an as-needed basis to the various data related classes. The work involved in this ticket was very detail oriented and involved reviewing hundreds of lines of C++ code located in tens to hundreds of classes depending on the specific aspect of the refactoring involved. As of mid-2019, the new *AAttributeSet* class has stabilized and the use of it by the subclasses in the hierarchy is a work in progress; with, perhaps, 80% to 90% coverage. Of

particular note is the use of 2 distinct types of attributes by one particular class in the hierarchy that will require a future change to this new class in order to support that functionality. For now, the specific customization is localized to the subclass and has become a TODO note for future consideration.

Ticket #218 – pull current attribute information out of EnSimDrawableObject. With the newly created *AAttributeSet* class, it became easier to localize current attribute selection information; further aiding developers and lowering future maintenance costs due to reducing the number of touch points required to understand how current attributes are maintained.

*Ticket #214 – pull *m_TimeAttribute out of CTableObject*. With the new *AAttributeSet* class implemented, it was fairly straightforward to remove the no-longer needed code from this class. For this work, a search of the 1,545 files revealed 32 possible locations that needed to be reviewed and adjusted appropriately.



The new colour scale showing both category and value.

Ticket #197 – CColourScale should display "oneof" category and not value. After improvements to multiple attributes a number of "ripple-effects" occurred; one of which was the incorrect display of a "*oneof*" value instead of its associated category.

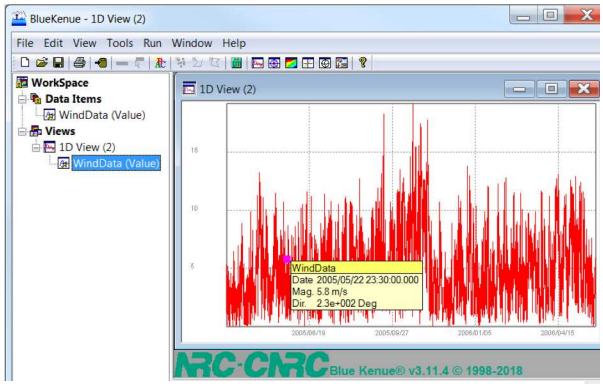
Ticket #236 – CColourScale dialog should display ABC=123 for "oneof" attributes. After improvements to multiple attributes it also became much easier to add the new feature of displaying both the category and the associated value; making Blue Kenue slightly easier to use operationally.

Ticket #249 – default colour scale does not work for ATR_INTEGER. For example, if an integer attribute had values 1 to 6 then 10 intervals could not be created; causing a fault. This ticket resolved that issue.

Ticket #186 – Modernize code that starts a thread. This work is related to making the source code easier for developers to use and enhance with new features. Historically, the set of lines of code to start a thread were all copy and pasted including the associated switch statement on all possible return states. Now a new function at the root of the class hierarchy has been added in order to localize the functionality in order to reduce future maintenance costs in terms of writing new code and also for easing understanding existing code by new developers.

Ticket #209 – pull AEnSimWorkspaceObject out of CEnSimObject. The work in this ticket will make it easier for developers to understand the coupling between the two aspects; now separated into two distinct classes. In the best case if the work was done well, a Blue Kenue user would not even know that something has changed with respect to how data objects interact with the work space. Further future enhancements in this area are planned.

Ticket #210 – pull AEnSimContextMenu out of CEnSimObject, CDecoObject, CStationsTableView and CEnSimView. Similar to #209, a Blue Kenue user would not be able to notice that this deep internal work had occurred. From a development point of view, this ticket removed redundant code in order to ease future maintenance and understanding.

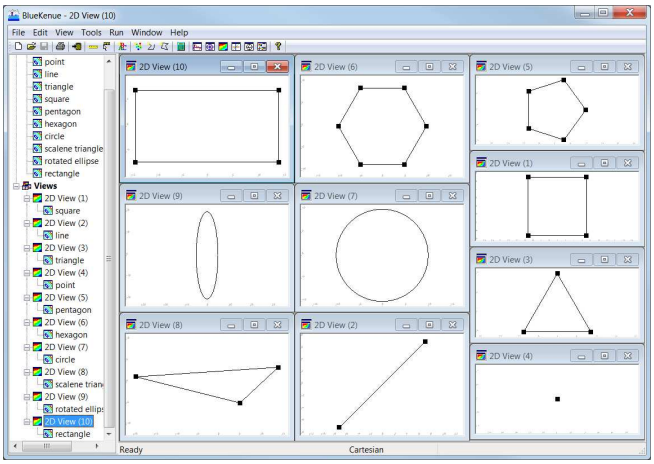


A popup for a point in a time series shown with values aligned

Ticket #259 – popup does not align numbers. When clicking on a point, line, cell, etc, the popup did not align the values in a column. This was resolved in multiple places.

Ticket #263 – attribute properties should also display attribute type in properties dialogs. 13 dialogs were modified in order to display each attribute's type (e.g. double) and to display both the category value and index (e.g. ABC=123). See also #236.

Ticket #275 – duplicated metadata is quietly overwritten. Metadata handling improvements are slowly starting to be made now that multiple attribute improvements have stabilized. In this case, when the same metadata is used more than once in a data file then a warning is displayed and the ability to select which value to use is presented.

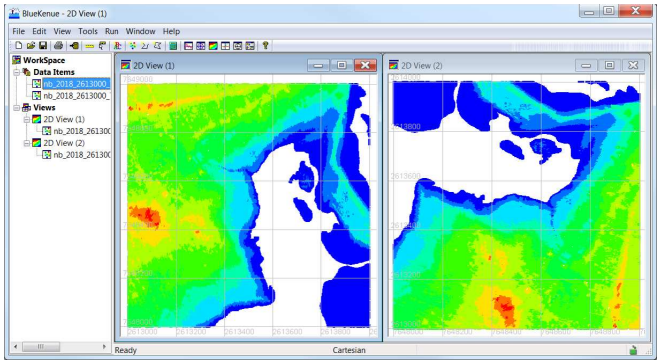


10 examples of creating simple new geometries in Blue Kenue (some are scaled and/or rotated) architecture

Ticket #243 – Add new functionality to create circle and rectangle 2-d line sets. The ticket's description proposed to add new functionality to be able to create a circle with a specified radius and origin. During review, the request was seen to be generalizable to points, lines, triangles, squares, pentagons, and, in general, scalable and rotatable n-gons (circles are n-gons with large n and ellipses are scaled circles).

F. Examples created in first half of 2019 and complete (12)

Ticket #337 – display time in a consistent user readable string. The time is presented in slightly differing ways throughout Blue Kenue. The work in the ticket was to find and resolve the differences. It was found that 34 different locations needed to become localized to one function; that was subsequently enhanced, thus propagating the enhancements to all 34 locations.



Swapping X and Y in an XYZ Point Set object (lat/lon vs. lon/lat)

Ticket #408 – Swap attributes in an XYZ Point Set. XYZ Files contain 3 columns of data. Different software applications create these files in different ways; some use space to delimit the columns, some use comma, some use semicolon etc. In addition, there may or may not exist a first line containing column names. Therefore, Blue Kenue has no way to determine if latitude (or longitude) will be in the first column (called X) or the second column (called Y). A new menu item was created (*Edit -> Point Set -> Swap X and Y*) to let the Blue Kenue user decide what should be done after an XYZ file has been loaded into the Workspace and inspected within a 2D View. Note that Blue Kenue has many types of point set files; here we added functionality specifically for the XYZ Point Set object containing 1 frame of data. General Point Sets or Parcel Sets possibly containing multiple data frames may be considered in the future. Also note that Blue Kenue has its own native XYZ Point Set file format that differs from other XYZ formats due to the addition of a more detailed header section before the data section of the file.

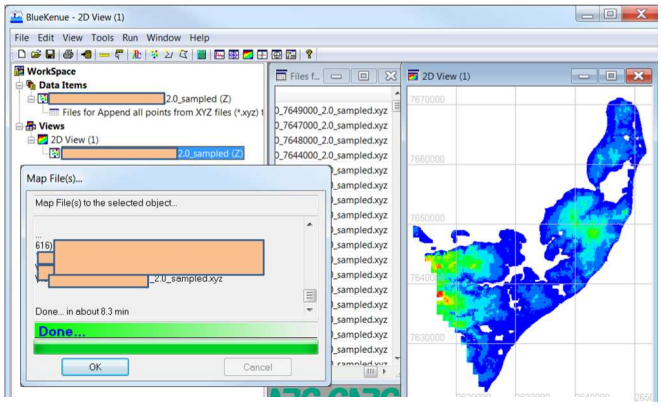
AVAILABLE BLUE KENUE REDUCTIONS FOR TOOLS -> MAP FILE(S)…

Selected Object in the Workspace	Available Reductions based on the Selected Object	File type that can be selected
Rect 2D Scalar	• Assign all DEMs • Sum all Rect 2D Scalars • Average all Rect 2D Scalars • Sum all grib files • Average all grib files • *More in Green Kenue*	• *.dem • *.r2s • *.r2s • *.grib, *.grib2 • *.grib, *.grib2 • **.fst, etc.*
XYZ Point Set	• Append all XYZ files	• *.xyz

Ticket #334 – Tools -> Map File(s). Is a larger piece of new functionality now available in Blue Kenue. It will likely take years before all possibilities are completed and made available to the community. The main idea of this new feature was inspired by the already existing Map Object functionality within Blue Kenue, where one object is selected in the workspace and then one other object is mapped onto it. One drawback of this approach happens when we have a large number of mappings that we need to perform because we need to load the files into the workspace and then map one object at a time onto the object of interest. Thus was born the idea of Map File(s) where we start by selecting one object in the workspace but then we don't map one loaded object but we instead directly map one or more files containing data. As a matter of terminology, since there may be more than one file that is being selected and since we are mapping down to one selected object, we can call this operation a "*reduction*". There are currently 6 available reduction methods (see table above) within Blue Kenue.

The first reduction method, interestingly, was motivated by a need within another application called Green Kenue – a hydrology oriented software that relies on a shared underlying technology with Blue Kenue. In that first motivating example, the idea was to summarize the mean precipitation over an area of interest as collected from about a year's worth of precipitation data (about 1,200 files stored in grib format). In this case, a Rect 2D Scalar was created via Green Kenue's grid generation functionality to cover the area of interest (all "data" was initially set to 0mm). Then the empty grid was selected in the workspace and the "Average all grib files" reduction was performed in order to summarize the years' worth of data in the grid.

An example relevant to pre-processing of data for input to TELEMAC using Blue Kenue has since been implemented. In this case, a user was interested in developing a digital elevation model combining multiple tiles of high resolution Light Detection and Ranging (LiDAR) data over a region of



631 XYZ files appended onto an XYZ Point Set object using Map File(s)
(In this case, each XYZ file is LiDAR topographic data [7])

interest in New Brunswick, on the east coast of Canada. The user collected 632 topographic files and converted each of them to XYZ format. Blue Kenue was used to load one of the XYZ files; which was then selected in the workspace and Map File(s) was used to append all of the other 631 files onto the selected XYZ Point Set object. The result was one XYZ Point Set object in the Blue Kenue workspace containing points from all 632 XYZ files. In addition, Blue Kenue did 2 more things; first it created a new table object containing a list of all files that were used during the reduction operation and added it as a child of the selected object in the workspace, and second, some new metadata was added onto the object in order to record which reduction operation was performed, approximately how long it took and how many files were selected by the user to map. This additional information may then be used in the future to help trace the origin of the processing and to support modelling quality assurance / control.

VII. Potential Future Blue Kenue Work

This section describes a few possible future activities related to Blue Kenue that may be of direct interest to the Blue Kenue community of users. Contributions, collaborations and/or contracts are very much welcome at any time.

A. Additional Reductions for Map File(s)…

More reduction methods are certainly possible, either as additions to the current set of reductions for the 2 previously mentioned objects, or as reductions on new objects depending on the needs of the Blue Kenue users. For example, merging multiple *.slf files (containing a single frame or multiple frames) may be of interest for some users. This may include preserving part of a model mesh and modifying another part and then merging them back together; or merging channel meshes with regular meshes, among other things. This would be a more direct methodology than the existing approach.

B. Blue Kenue Reference Manual Migration and Update

The current manual [1] was last modified about 8 years ago and used software call FrameMaker [4] for its production. The last portion of 2019 will be used to migrate the existing information in this system to a new documentation preparation system called LaTeX [5]. These new files will be managed through CHyMS [3]. Once the migration process has completed then the LaTeX files will be updated to bring them closer to the current Blue Kenue features. Such placement under version control (in CHyMS) may help to more easily keep the information aligned in the future.

For example, Blue Kenue Community members interested in participating in keeping the user manual updated could request access to CHyMS (in 2020) and then request changes to the LaTeX files by submitting a patch file. The patch file would then be reviewed and considered to be applied. Examples of changes via patch files could be: (i) to fix small typos, (ii) to correct processes, or (iii) to add large paragraphs of helpful background and/or clarifying information. The idea is for the patches to be curated and for those that pass inspection to be applied to the reference manual. Proper acknowledgement would also be included in the subsequent Blue Kenue release.

C. Micro-plastic particles in the Marine Environment

Modern numerical models can be effective in predicting the movement and fate of particles in oceans and waterways, and are routinely applied in other water quality contexts. Blue Kenue is proposed to aid research (where appropriate) in: (i) developing techniques for simulating the behaviour and transport of micro-plastics across a broad range of spatial and temporal scales of relevance; (ii) characterizing particle properties and degradation processes in diverse aquatic environments; (iii) helping to fill gaps in the availability of field observations for input to, and verification of numerical models; and (iv) suitability for other challenges in marine and freshwater environments such as debris transport and accumulation, ice jams and related flooding in rivers, oil spill simulation and emergency response, ecological or agent-based modelling

For example, Blue Kenue enhancements may include new components for: (i) analysis of remote sensing data to estimate micro-plastic concentrations in waterbodies; (ii) integration of numerical simulation results of hydrodynamics in lakes, rivers, estuaries and oceans, and (iii) prediction of micro-plastic transport, dispersion, degradation and accumulation.

D. Tool(s) to Ease Preparation of Environmental Inputs

An ongoing new addition to TELEMAC2D is a river ice module named KHIONE. This module is to simulate main river ice processes including frazil ice formation and transport, border and dynamic ice and ice jam and its impact on river surface elevation and potential flooding. Main inputs to KHIONE in addition to usual inputs to open water version of TELEMAC2D are air and sky conditions as well as initial river temperature and frazil concentration. Blue Kenue can currently visualize outputs of the existing version of KHIONE. It is, however, worth exploring how and if Blue Kenue can be further developed to facilitate pre- and post-processing for TELEMAC2D river ice simulations.

For example, one possibility could be the development of tools to ease the preparation of format compatible environmental inputs. This includes automatic conversion of main data formats of climate models providing data such as

cloud coverage, wind, precipitation, air temperature and humidity.

Conclusions

CHyMS is important within the context of possible future Blue Kenue developments because it allows more than one developer to work on the code at a time and also helps transition (i.e. handing over) of code from one person to another during times of change. CHyMS is also very beneficial from the point of view of a communication mechanism between developers and users and/or to management and/or other sources of funding for new features and maintenance activities. CHyMS significantly helps to organize and stay on top of the most pressing issues facing Blue Kenue, Green Kenue, pyEnSim and ECDE along with continuing to keep historical applications such as AnemoScope, MarKE and others up-to-date with the latest improvements to the shared underlying technology.

Acknowledgements

All OpenTelemac forum users are thanked for taking their time to carefully explain any Blue Kenue issues they have encountered. They are also thanked for their patience. The NRC-OCRE is thanked for their sponsorship of this work over the past years. All NRC-OCRE past and present employees are thanked for their helpful feedback present in many CHyMS tickets. Martin Serrer is thanked for all of the work he has put into Blue Kenue over the years. I hope that he is now enjoying his well-earned retirement. Ivana Vouk is thanked for the need of new Green Kenue functionality that led to my design of the new Map File(s) functionality. Sean Ferguson is thanked for sharing 632 LiDAR datasets in XYZ format with me. Parishat Tanakoor is thanked for his excellent work migrating and helping to update the Blue and Green Kenue User Manuals that will make their way into future installers. Vahid Pilechi is thanked for this ideas in the section on possible Blue Kenue and micro-plastics related future directions and Hossein Babaei is thanked for his ideas in the section related to Blue Kenue and KHIONE. The anonymous reviewer is thanked for their time spent reading and commenting on the first submitted version of this paper. And lastly, Enda Murphy is thanked for his insightful internal review comments for both the initial and final submissions of this paper.

References

[1] Canadian Hydraulics Centre, National Research Council Canada (September 2011). "*Blue Kenue Reference Manual*"

[2] TELEMAC-MASCARET consortium (August 2019). "*open TELEMAC-MASCARET – The mathematically superior suite of solvers – forum*". Retrieved from http://www.opentelemac.org

[3] Ocean, Coastal and River Engineering, National Research Council Canada (August 2019). "*Canadian Hydrological Model Stewardship Frequently Asked Questions"*. Retrieved from https://chyms.nrc.gc.ca

[4] Wikipedia® (August 2019). "Adobe FrameMaker". Retrieved from https://en.wikipedia.org/wiki/Adobe_FrameMaker

[5] LaTeX3 Project (August 2019). "The LaTeX Project". Retrieved from https://www.latex-project.org/

[6] National Research Council Canada (August 2019), "*Blue Kenue: software tool for hydraulic modellers*". Retrieved from https://nrc.canada.ca

[7] Province of New Brunswick, Canada (October 2019), "*GeoNB Data Catalogue*". From http://www.snb.ca/geonb1/e/DC/catalogue-E.asp